

1/1



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 06337762

(43)Date of publication of application: 06.12.1994

(51)Int.Cl.

G06F 3/06
G06F 12/00

(21)Application number: 05126734

(71)Applicant:

HITACHI LTD
HITACHI SOFTWARE ENG CO
LTD

(22)Date of filing: 28.05.1993

(72)Inventor:

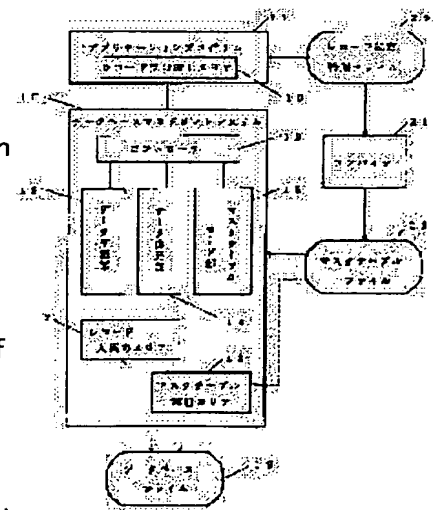
GOMI FUMIO
KAWASAKI YOSHIFUMI
HORIE MASAHISA
TOYAMA YUJI
IWATO TAKENORI
NOGAMI YOSHIFUMI

(54) COMPRESSING METHOD AND RESTORING METHOD FOR DATA BASE RECORD

(57)Abstract:

PURPOSE: To shorten the length of data base records and reduce the capacity by compressing data of all initial value data item parts when there are many initial value data items in the data base records.

CONSTITUTION: When record data set in a record receiving and passing area 16 secured in an application program 11 are exclusively 0Red by using a mask table, all data item parts having the same values between the record data and mask table are subtracted to become zero and then the contents of masked record data in the record receiving and passing area 16 are left. On the basis of the masked record data, compression control information is substituted for parts of successive identical characters, noncompression control information is set even at the head of discontinuous parts, and compressed record data are set in a record input/output area 17 and stored in a data base file 19.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平6-337762

(43) 公開日 平成6年(1994)12月6日

(51) Int.Cl.⁵

G 0 6 F 3/06
12/00

識別記号

3 0 1 W
5 1 1

庁内整理番号

8944-5B

F I

技術表示箇所

審査請求 未請求 請求項の数 3 O L (全 12 頁)

(21) 出願番号

特願平5-126734

(22) 出願日

平成5年(1993)5月28日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(71) 出願人 000233055

日立ソフトウェアエンジニアリング株式会
社

神奈川県横浜市中区尾上町6丁目81番地

(72) 発明者 五味 文男

神奈川県横浜市戸塚区戸塚町5030番地 株
式会社日立製作所ソフトウェア開発本部内

(74) 代理人 弁理士 秋田 収喜

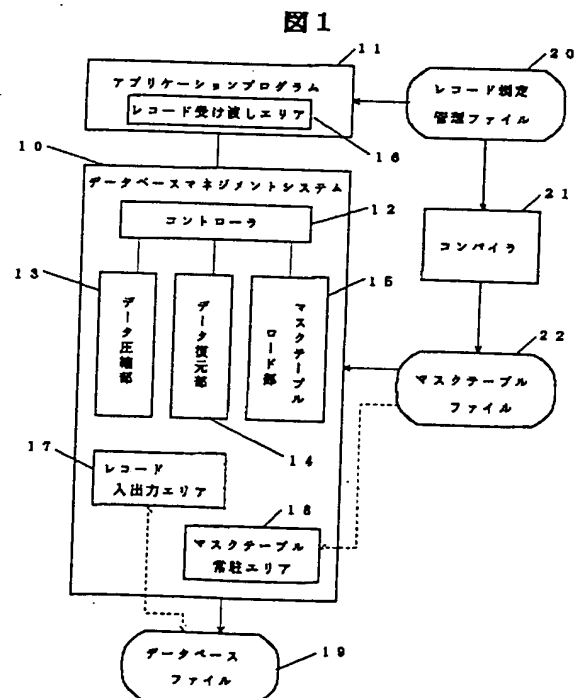
最終頁に続く

(54) 【発明の名称】 データベースレコードの圧縮方法および復元方法

(57) 【要約】

【目的】 データベースレコードのデータにおいて同一文字の連続性がとぎれる場合でもデータベースレコードの圧縮効率を上げ、データベースファイルへの格納するデータベースレコード長を短くする。

【構成】 レコード形式単位に各データ形式に応じた初期値を設定したマスクテーブルファイル22を作成しておき、格納要求されたレコード受渡しエリア16とマスクテーブル常駐エリア18との排他的論理和をとり、値が一致する項目全てをゼロ"00"にし、同一文字の連続性を大幅に高めた後、連続同一文字の部分を圧縮制御情報に置換し、データベースファイル19に格納する。



【特許請求の範囲】

【請求項1】 データベースマネジメントシステムにおけるデータベースレコードの圧縮方法であって、レコードフォーマットが固定のデータベースレコードに対して、前記データベースレコード内のデータ形式に応じた初期値で設定されたマスクテーブルを作成して、1データベースレコード単位に格納されるデータベースレコードのデータと前記マスクテーブルとの排他的論理和をとり、前記データベースレコード内の初期値項目部分に対応するビットをオフにして連続同一文字列を作り、前記データベースレコードのデータにおける連続同一文字列がある位置に連続同一文字列に関する情報をもつ圧縮情報を置換し、前記データベースレコードのデータをビットオフにならなかった残りのデータベースレコードのデータと圧縮情報とからなるデータベースレコードのデータに圧縮することを特徴としたデータベースレコードの圧縮方法。

【請求項2】 請求項1に記載のデータベースレコードの圧縮方法によって、圧縮されたデータベースレコードが格納されている外部記憶装置上のデータベースファイルから圧縮されたデータベースレコードを読み出し、このデータベースファイルから読み出したデータベースレコード内の圧縮情報に置換された位置に圧縮情報をもとに連続同一文字を置換し、データベースレコードのデータ圧縮時と同一のマスクテーブルとの排他的論理和をとり対応するビットをオンにし、データ圧縮前のデータベースレコードに復元することを特徴としたデータベースレコードの復元方法。

【請求項3】 請求項1、または請求項2に記載のデータベースレコードの圧縮方法または復元方法において、前記マスクテーブルの作成は、データベースレコードを処理するアプリケーションプログラムの作成で使用するレコードフォーマット規定部分のソースプログラムをそのまま使用し、言語に応じた初期値データをもとに行われることを特徴としたデータベースレコードの圧縮方法または復元方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、主にデータベースマネジメントシステムにおけるデータベースレコードの圧縮方法および復元方法に関するものである。

【0002】

【従来の技術】 従来のデータ圧縮方法では、データベースレコード内に同一文字が3バイト以上連続している場合には、繰返文字数をカウントし、文字種別と繰返回数をそれぞれ各1バイトずつ合計2バイトの圧縮情報としてレコード内の同一文字があった位置に置換し、データベースレコードを圧縮している。

【0003】 この種に関連するものには例えば特開平4-348617号公報等が挙げられる。

【0004】

【発明が解決しようとする課題】 しかし、上記従来技術では、データベースレコード内に同一文字が3バイト以上連続している場合のみデータ圧縮の対象となっているため、データベースレコード内の初期値データ項目内に4ビットの符号コードが付くことにより同一文字の連続性がとぎれるような場合には、圧縮効率が上がらないという問題があった。

【0005】 本発明の目的は、データベースレコードのデータにおいて、同一文字の連続性がとぎれる場合でもデータベースレコードの圧縮効率を上げ、データベースファイルへの格納するデータベースレコード長を短くすることである。

【0006】

【課題を解決するための手段】 上記目的を達成するために、本発明の第1の手段は、データベースレコード内のデータ形式に応じた初期値で設定されたマスクテーブルを作成して、1データベースレコード単位に格納されるデータベースレコードのデータと前記マスクテーブルとの排他的論理和をとり、前記データベースレコード内の初期値項目部分に対応するビットをオフにして連続同一文字列を作り、前記データベースレコードのデータにおける連続同一文字列がある位置に連続同一文字列に関する情報をもつ圧縮情報を置換し、前記レコードデータをビットオフにならなかった残りのデータベースレコードのデータと圧縮情報とからなるデータベースレコードのデータに圧縮することを特徴とする。

【0007】 本発明の第2の手段は、外部記憶装置上のデータベースファイルに一旦格納されている圧縮されたデータベースレコードを読み出し、このデータベースファイルから読み出したデータベースレコード内の圧縮情報に置換された部分を連続同一文字の長さ分に拡張し、データベースレコードのデータ圧縮時と同一のマスクテーブルとの排他的論理和をとり対応するビットをオンにし、データベースレコードのデータ圧縮前のデータベースレコードに復元することを特徴とする。

【0008】 前記マスクテーブルの作成は、データベースレコードを処理するアプリケーションプログラムの作成で使用するレコードフォーマット規定部分のソースプログラムをそのまま使用し、言語に応じた初期値データをもとに行うことを特徴とする。

【0009】

【作用】 前述の手段によれば、データベースレコード内のデータ形式に応じた初期値、または発生頻度の高い任意の文字列で設定されたマスクテーブルを作成して、1データベースレコード単位に格納されるデータベースレコードのデータと前記マスクテーブルとの排他的論理和をとり、前記データベースレコード内の初期値項目部分に対応するビットをオフにして連続同一文字列を作り、前記データベースレコードのデータにおける連続同一文

3

字列がある位置に連続同一文字列に関する情報をもつ圧縮情報を置換し、前記レコードデータをビットオフにならなかった残りのデータベースレコードのデータと圧縮情報とからなるデータベースレコードのデータに圧縮するので、圧縮可能な範囲が従来より広くとれ、レコードデータの圧縮効率が向上し、データベースファイルへの格納データベースレコード長を短くすることが可能となる。

【0010】そして、圧縮前のデータベースレコードに復元する時は、外部記憶装置上のデータベースファイルに一旦格納されている圧縮されたデータベースレコードを読み出し、データベースファイルから読み出したデータベースレコード内の圧縮情報に置換された部分を連続同一文字の長さ分に拡張し、データベースレコードのデータ圧縮時と同一のマスクテーブルとの排他的論理和をとり対応するビットをオンにし、データベースレコードのデータ圧縮前のデータベースレコードに簡単に復元することが可能である。

【0011】更に、マスクテーブルの作成は、データベースレコードを処理するアプリケーションプログラムの作成で使用するレコードフォーマット規定部分のソースプログラムをそのまま使用し、言語に応じた初期値データをもとに行われるため、データベースレコードを処理するアプリケーションプログラムとデータベースファイルのズレを防止できる。

【0012】

【実施例】以下、図面を使用して本発明の一実施例について詳細に説明する。

【0013】図1は、本発明を実現するデータベースマネジメントシステムの全体構成を示すブロック図であり、図2は、図1のデータベースマネジメントシステムを実施するための装置のハードウェア構成を示すブロック図である。

【0014】図1および図2において、10はデータベースマネジメントシステム、11はアプリケーションプログラム、19はデータベースファイル、20はレコード規定管理ファイル、21はコンパイラ、22はマスクテーブルファイル、31は入出力装置、32は中央処理装置(CPU)、33は主記憶装置、34は外部記憶装置である。

【0015】図1に示すように、本発明の実施例のデータ圧縮方法を実行するデータベースマネジメントシステム10は、アプリケーションプログラム11が使用するデータベースファイル19のデータベースレコードの入出力要求により動作し、データベースマネジメントシステム内を制御するコントローラ12、データベースレコードの圧縮処理を行うデータ圧縮部13、データベースレコードの復元処理を行うデータ復元部14、マスクテーブルファイル22の入力を行うマスクテーブルロード部15から構成される。

4

【0016】また、使用するマスクテーブルファイル22は、レコード規定管理ファイル20をもとにコンパイラ21により作成され、ファイル内容はデータベースマネジメントシステム10の開始時に、マスクテーブルロード部15によってマスクテーブル常駐エリア18に常駐化される。

【0017】図3は、データベースレコードのレコードフォーマット、およびデータ形式を規定するレコード規定管理ファイル20に設定されているレコード形式の一例をもとに、コンパイラ21に入力し、オブジェクト形式データに変換後、マスクテーブル23がマスクテーブルファイル22に作成される手順を示したものである。

【0018】この図3の例では、コンパイラにより内部10進形式(バック形式10進データ)20aに対しては00...0Cの形式で、外部10進形式(ゾーン形式10進データ)20bに対しては、F0F0...の形式で、文字形式(文字列データ)20cに対しては、4040...の形式で、漢字形式(日本語列データ)20dに対しては、A1A1...の形式でマスクテーブルは設定される。

【0019】このようにして、データベースレコードのレコードフォーマット、およびデータ形式を規定するレコード規定管理ファイル20に設定されているレコード形式をコンパイラに入力することにより簡単にマスクテーブルを作成できる。

【0020】また、データベースレコードを処理するアプリケーションプログラムの作成で使用するレコードフォーマット規定部分のソースプログラムをそのまま使用し、言語に応じた初期値を設定することにより、データベースレコードを処理するアプリケーションプログラムとデータベースファイルのズレを防止できる。

【0021】図4は、アプリケーションプログラム11から格納要求されたデータベースレコード(以下、レコードと略す)の一例を用いて、データ圧縮する処理過程を説明するためのものである。

【0022】アプリケーションプログラム11内に確保されたレコード受け渡しエリア16内にセットされたレコードデータ16aに対して、マスクテーブル23で排他的論理和をとると、レコードデータ16aとマスクテーブル23間で同一の値をもつ全てのデータ項目部分が減算されゼロの値となり、結果はレコード受渡しエリア16のマスク済みレコードデータ16bの内容となる。

【0023】そして、そのマスク済みレコードデータ16bをもとに、連続した同一文字の部分の圧縮制御情報(文字と長さ)に置換し、また不連続部分の先頭にも非圧縮制御情報(長さ)を設定し、レコード入出力エリア17に圧縮レコードデータ17aをセットし、その圧縮レコードデータ17aをデータベースファイル19へ格納する。

【0024】次に、図5は、図4の説明で使したレコ

ードの一例を用いて、データベースファイル19に格納された圧縮レコードデータ17aを読み出し、データ圧縮部分を元のレコードデータ16aの内容に復元し、アプリケーションプログラム11に渡す処理過程を説明するためのものである。

【0025】データベースファイル19から読み出されたレコード入出力エリア17内の圧縮レコードデータ17aの圧縮制御情報と非圧縮制御情報をもとに、連続同一文字圧縮の場合は、圧縮制御情報内の文字をその長さ分に拡張し、それ以外の場合は非圧縮制御情報内の長さ分の非圧縮データ部分そのままを、レコード受け渡しエリア16にセットする。

【0026】すなわち、レコード受渡しエリア16に圧縮レコードデータ17aを圧縮制御情報および非圧縮制御情報より変換した変換レコードデータ16cとしてセットする。なお、このときの変換レコードデータ16cはマスク済みレコードデータ16bと同一のデータ内容となる。

【0027】そして、その変換レコードデータ16cに対して、格納時に使用したマスクテーブル23で排他的論理和をとると、変換レコードデータ16cのゼロの値の全ての部分にマスクテーブルの値が加算され、結果はレコード受渡しエリア16の復元レコードデータ16dに示すように圧縮前のレコードデータ16aの内容に復元され、その復元レコードデータ16dをアプリケーションプログラム11へ渡す。

【0028】図6は、本実施例のデータ圧縮の処理手順を示したフローチャートであり、図4をもとにして説明する。

【0029】アプリケーションプログラム11から格納要求されたレコード受渡しエリア16のレコードデータ16aの内容をもとに対応するマスクテーブル23を決定する(ステップ100)。ここで決定したマスクテーブル23を用いてレコード受け渡しエリア16のレコードデータ16aの内容に対し、1命令で256バイトづつの排他的論理和を実行する(ステップ110)。

【0030】この実行結果のレコード受け渡しエリア16のレコードデータ16bをもとに、1命令で256バイトづつ連続同一文字を検索(ステップ120)、2バイト以上の連続同一文字を検出した場合は(ステップ130)、同一文字の連続する部分の長さを求め(ステップ140)この文字と長さをもとに圧縮制御情報を作成し、レコード入出力エリア17にセット(ステップ150)する。

【0031】連続同一文字がない場合は(ステップ130)、不連続文字部分の長さをもとに非圧縮制御情報を作成し、非圧縮データ部分をレコード入出力エリア17にセットする(ステップ160)。

【0032】次に、ステップ120からステップ160までをマスク済みレコードデータ16bの検索が終了す

るまで繰返し、検索終了時には、レコード入出力エリアの圧縮レコードデータ17aをセットし、その内容をデータベースファイル19へ格納(ステップ170)し要求元へリターンする。

【0033】図7は、本実施例のデータ復元の処理手順を示したフローチャートであり、図5をもとにして説明する。

【0034】アプリケーションプログラム11からのデータベースレコード読み出し要求をもとに、データベースファイル19から圧縮レコードデータ17aをレコード入出力エリア17へ読み出す(ステップ200)、この内容をもとに対応するマスクテーブルを決定する(ステップ210)、レコード入出力エリア17内の圧縮レコードデータ17aの先頭から圧縮制御情報、または非圧縮制御情報を検索し(ステップ220)、圧縮制御情報の場合は(ステップ230)、圧縮制御情報内の文字をその長さ分に拡張し、レコード受渡しエリア16へセットする(ステップ240)。

【0035】非圧縮制御情報の場合は(ステップ230)、非圧縮制御情報内の長さ分の非圧縮データ部分をそのままレコード受渡しエリア16へセットする。

【0036】ステップ220からステップ250までをレコード入出力エリア17の圧縮レコードデータ17aが終了まで繰返し、検索終了時には、レコード受渡しエリア16の変換レコードデータ16cに対して先に決定したマスクテーブル23を用いて、1命令で256バイトづつの排他的論理和を実行する(ステップ260)。この実行結果は、レコード受渡しエリア16の復元レコードデータ16dとなる。

【0037】そして、この復元レコードデータ16dは、元のレコードデータ16aと同一のものであり、要求元へリターン時、この復元レコードデータ16dが要求元へ渡される。

【0038】以上説明したように、本発明によれば、レコード件数が多く、レコード内のデータ項目数が多い大規模データベースシステムにおいて、データベースレコード内に内部10進形式等の初期値データ項目が多数存在している場合は、これら全てのデータ項目部分をデータ圧縮するため、データベースレコード長が大幅に短縮され、データベースファイルを格納する外部記憶装置容量の大幅な削減となる。

【0039】このため、データベースレコード格納時と読み出し時のデータ転送時間も大幅な削減となる。

【0040】そして、圧縮されたデータベースレコードから圧縮される前のデータベースレコードに復元する場合も、レコード圧縮で行った処理手順を逆にすることで簡単に復元できる。

【0041】また、データベースレコードを処理するアプリケーションプログラムの作成で使用するレコードフォーマット規定部分のソースプログラムをそのまま使用

7

し、言語に応じた初期値を設定することにより、データベースレコードを処理するアプリケーションプログラムとデータベースファイルのズレを防止できる。

【0042】

【発明の効果】以上説明したように、本発明によれば、レコード件数が多く、レコード内のデータ項目数が多い大規模データベースシステムにおいて、データベースレコード内に内部10進形式等の初期値データ項目が多数存在している場合は、これら全てのデータ項目部分をデータ圧縮するため、データベースレコード長が大幅に短縮され、データベースファイルを格納する外部記憶装置容量の大幅な削減となる。

【0043】さらにデータベースレコード格納時と読み出し時のデータ転送時間も大幅な削減となる。

【図面の簡単な説明】

【図1】本発明を実現するデータベースマネジメントシステムの全体構成を示すブロック図である。

【図2】本発明を実現するハードウェアの構成を示すブロック図である。

【図3】本実施例のマスクテーブル作成手順を説明するための図である。

【図4】本実施例のデータ圧縮する処理過程を説明するための図である。

【図5】本実施例のデータ復元する処理過程を説明するための図である。

【図6】本実施例のデータ圧縮の処理手順を示すフローチャートである。

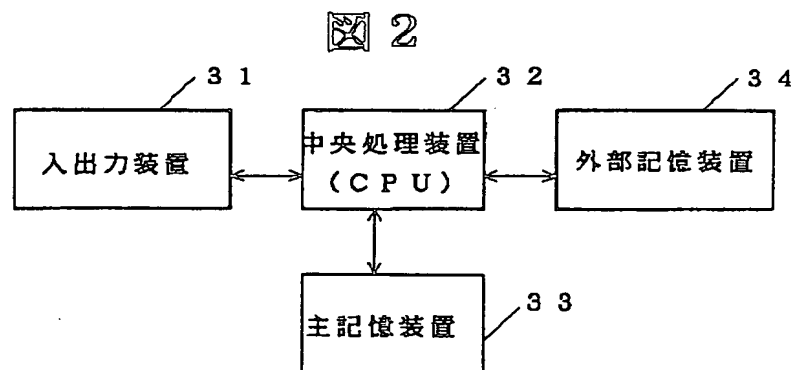
8

【図7】本実施例のデータ復元の処理手順を示すフローチャートである。

【符号の説明】

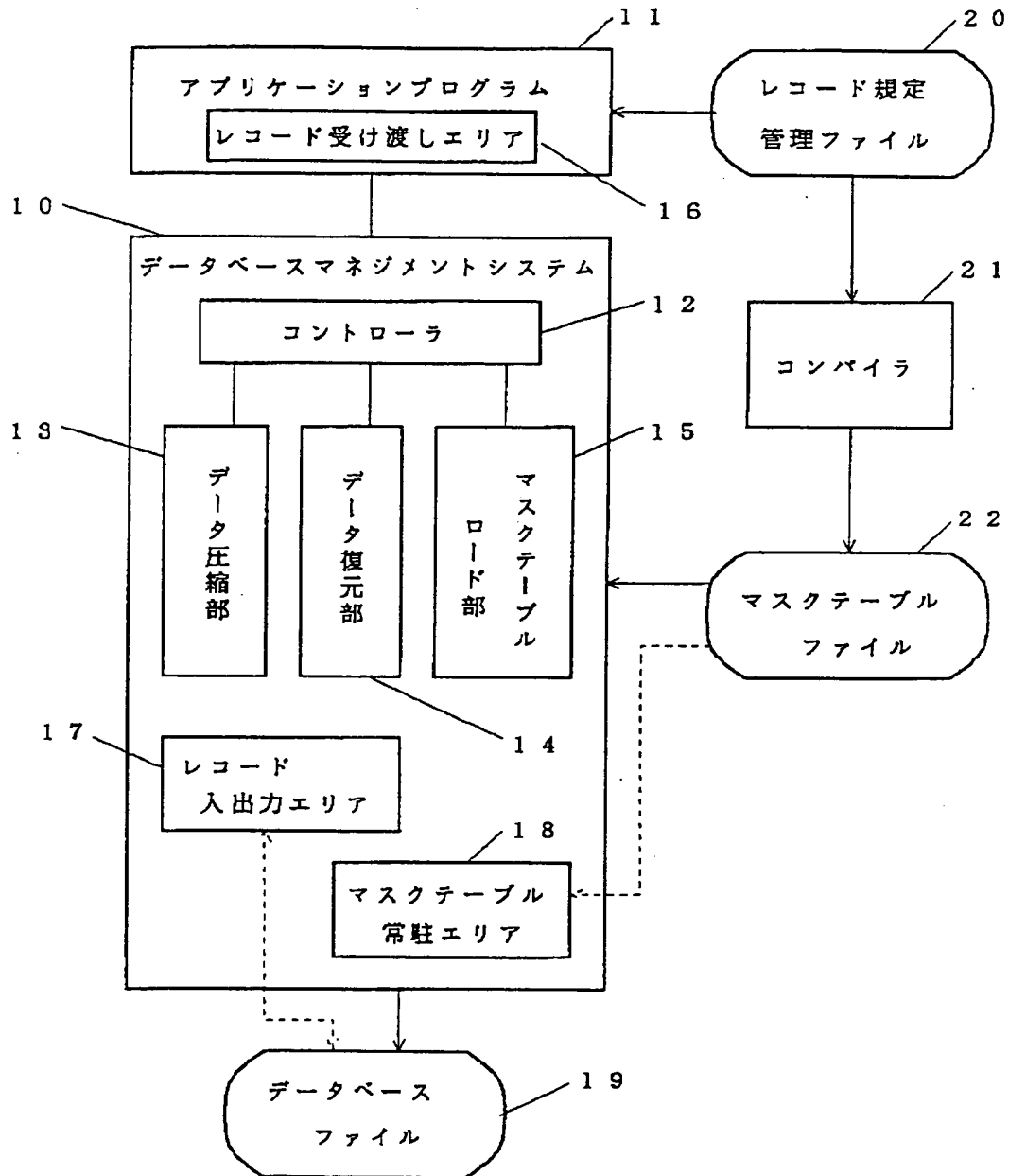
- 10 …データベースマネジメントシステム、
- 11 …アプリケーションプログラム、
- 12 …コントローラ、
- 13 …データ圧縮部、
- 14 …データ復元部、
- 15 …マスクテーブルロード部、
- 16 …レコード受け渡しエリア、
- 16 a …レコードデータ、
- 16 b …マスク済みレコードデータ、
- 16 c …変換レコードデータ、
- 16 d …復元レコードデータ、
- 17 …レコード入出力エリア、
- 17 a …圧縮レコードデータ、
- 18 …マスクテーブル常駐エリア、
- 19 …データベースファイル、
- 20 …レコード規定管理ファイル、
- 21 …コンパイラ、
- 22 …マスクテーブルファイル、
- 23 …マスクテーブル。
- 31 …入出力装置
- 32 …中央処理装置
- 33 …主記憶装置
- 34 …外部記憶装置

【図2】



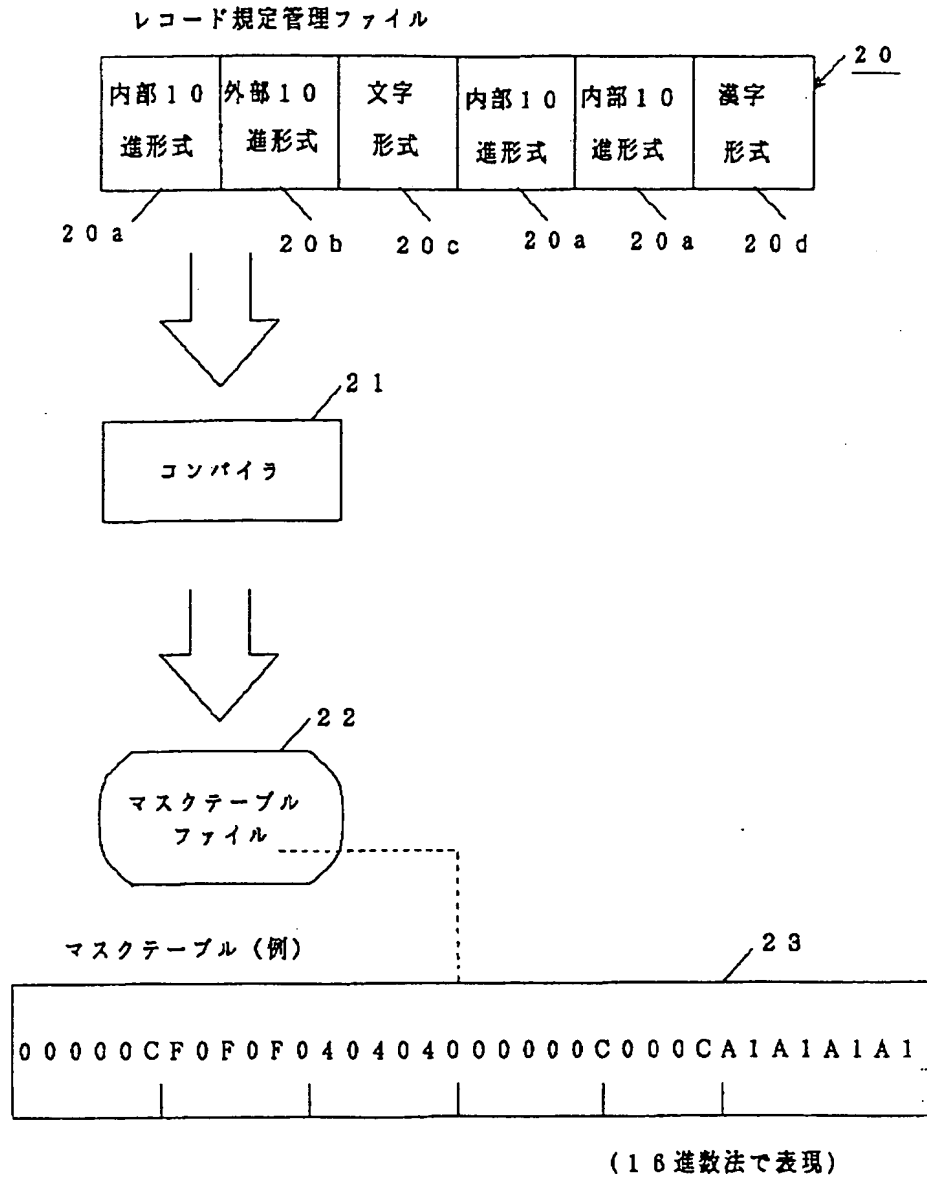
【図1】

図 1



【図3】

図3



【図4】

図 4

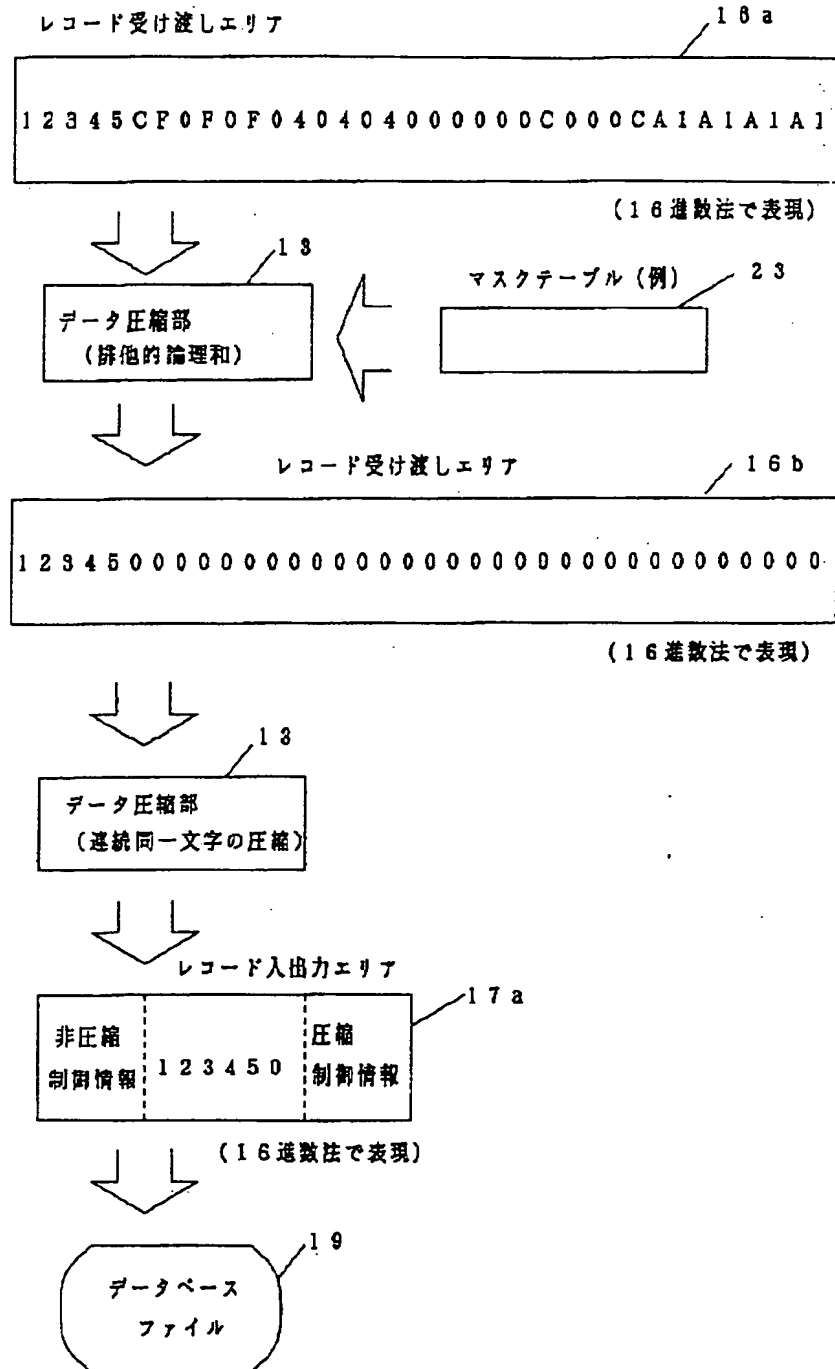
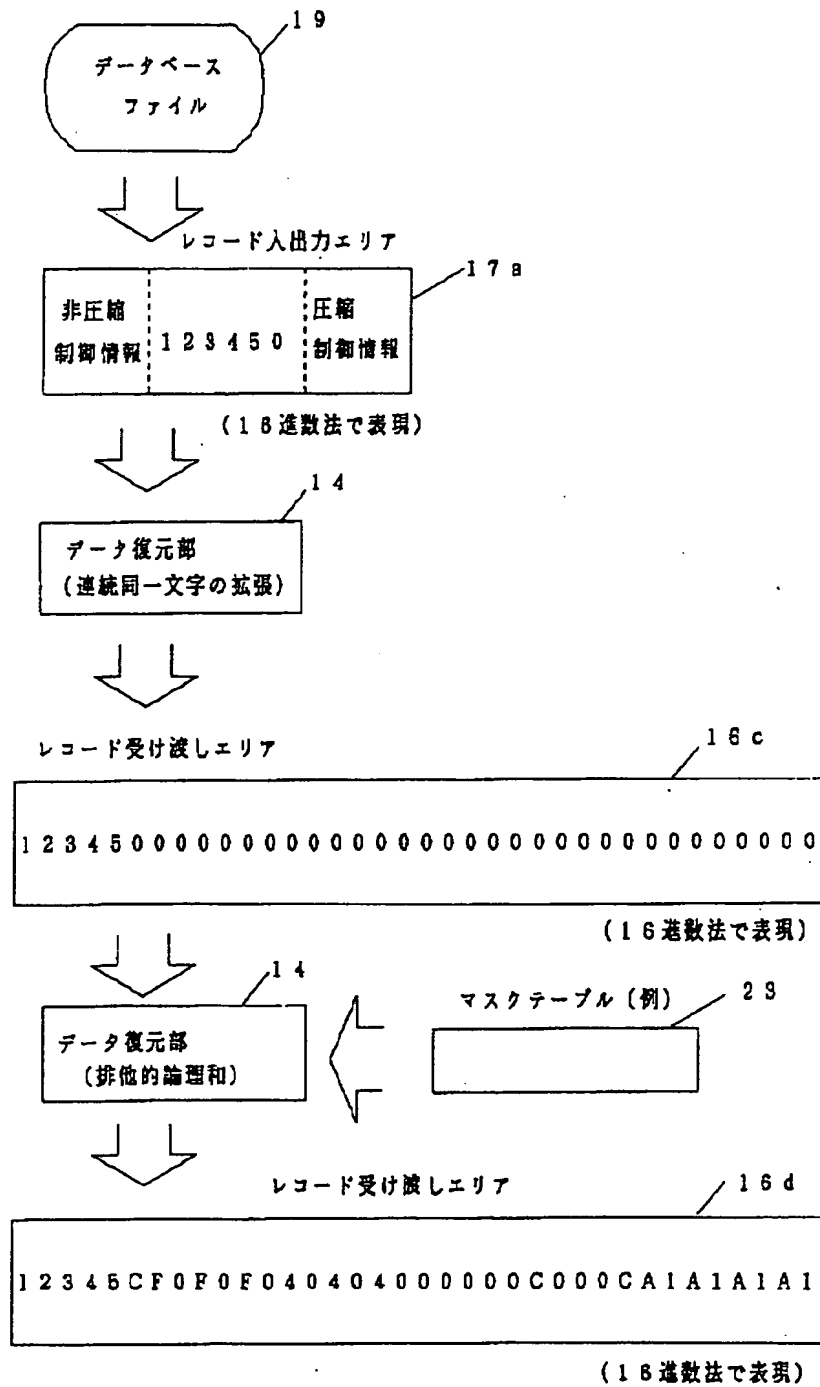
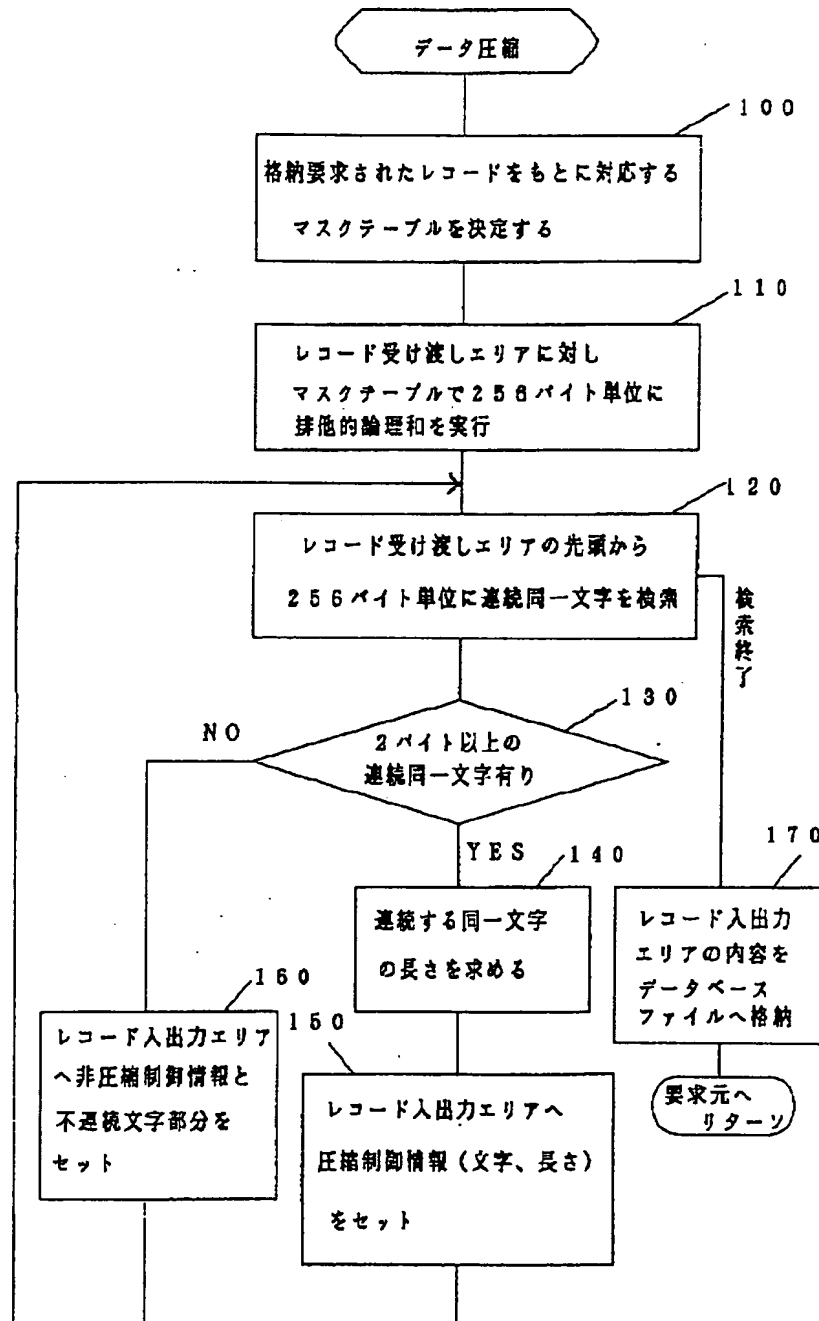


图 5



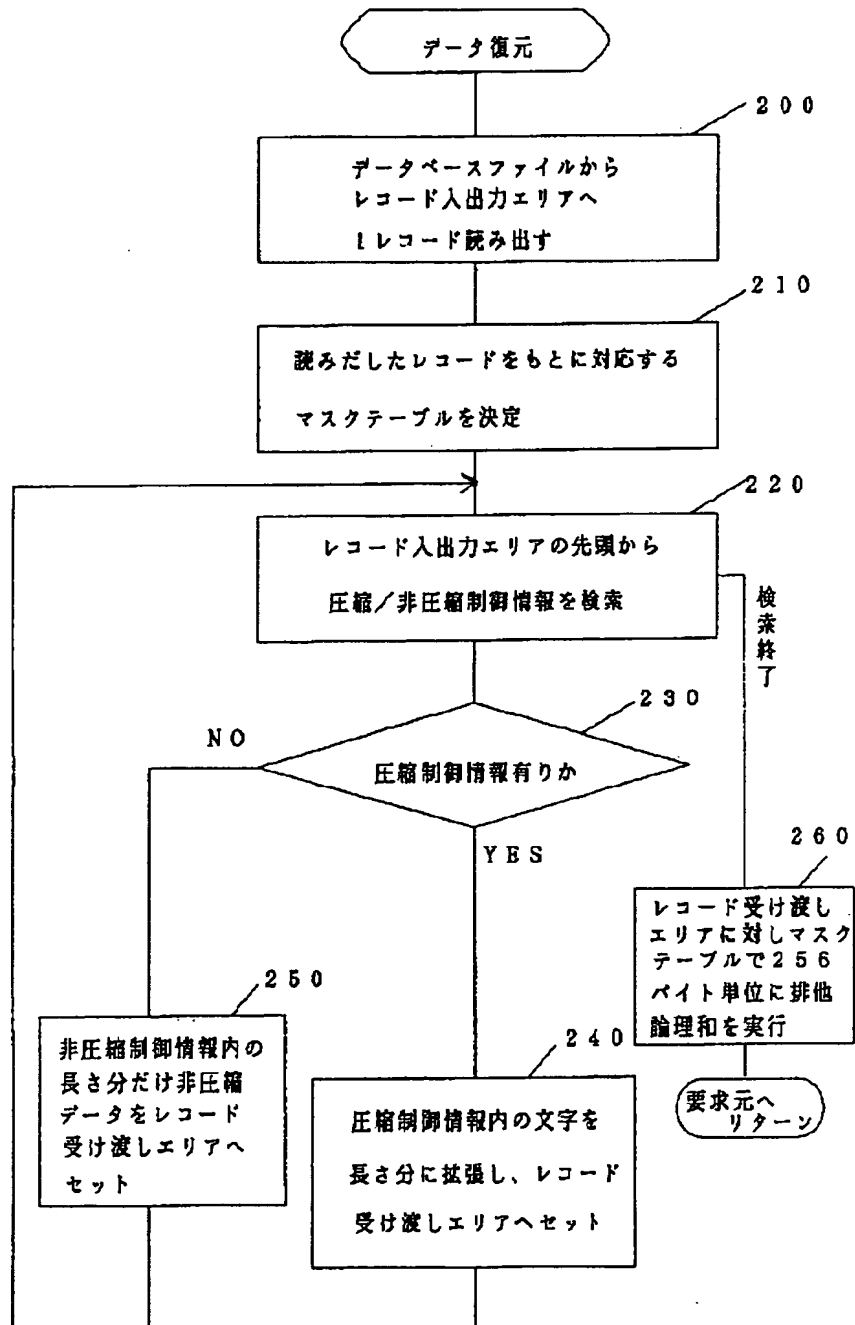
【図6】

図6



【図7】

図7



フロントページの続き

(72) 発明者 川崎 良文
神奈川県横浜市中区尾上町 6 丁目 81 番地
日立ソフトウェアエンジニアリング株式会
社内
(72) 発明者 堀江 正久
神奈川県横浜市中区尾上町 6 丁目 81 番地
日立ソフトウェアエンジニアリング株式会
社内

(72) 発明者 外山 雄司
神奈川県横浜市中区尾上町 6 丁目 81 番地
日立ソフトウェアエンジニアリング株式会
社内
(72) 発明者 岩戸 丈典
神奈川県横浜市戸塚区戸塚町 5030 番地 株
式会社日立製作所ソフトウェア開発本部内
(72) 発明者 野上 敬文
神奈川県横浜市戸塚区戸塚町 5030 番地 株
式会社日立製作所ソフトウェア開発本部内